The final version of this paper is available at https://ieeexplore.ieee.org/document/8516476
DOI: 110.1109/CBMI.2018.8516476

Temporal Compression and Fast Matching of Hand-crafted and Deep Features of Video Segments

Werner Bailer and Stefanie Wechtitsch
DIGITAL – Institute for Information and Communication Technologies
JOANNEUM RESEARCH Forschungsgesellschaft mbH
Steyrergasse 17, 8010 Graz, Austria
Email: {firstname.lastname}@joanneum.at

***Abstract*—In order to enable efficient instance search in video, compact descriptors for video segments have been proposed. They exploit the temporal redundancy within a video segment to obtain smaller descriptors, and the segment-based representation can be exploited to enable more efficient matching. In this paper we analyze the performance of different visual features when applying both lossless and lossy compression to the set of descriptors of one video segment. We consider both hand-crafted and deep features, i.e., visual features obtained from training a deep convolutional neural network. We also propose optimizations to the extraction and matching procedure. Both the compression methods and the optimizations are experimentally evaluated on a large video data set.**

I. Introduction

Many applications require identifying similar content in video beyond copy detection, e.g., finding video segments that show similar locations or salient objects. This requires robust and efficient methods for instance search in video. In application scenarios involving live streams, e.g., broadcasting applications or user generated video, feature extraction and matching need to provide results with low latency. This requires that the visual feature extraction and matching process is performed very efficiently. The visual similarity between streams may be partial, views may differ significantly and occlusions are likely to occur. The matching method must thus be able to handle these issues, i.e., go well beyond the variations supported by fingerprinting and copy detection methods. It is important that the method takes the temporal dimension of the video into account, i.e., is able to return a similarity score that considers the entire segment in which matches are found. This is not only an issue of the granularity of results, but also of efficiency, i.e., being capable of matching video segments rather than performing pairwise matching of sequences of key frames.

Using a framework for compact video descriptors [1], we analyze the performance of different local and aggregated visual features. This includes established hand-crafted features as well as deep features, i.e., feature representations that are obtained from trained deep convolutional neural networks. The first contribution of this paper is the analysis of the effect of lossless and lossy compression along the temporal dimension on the descriptor size, and (in the case of lossy compression) on the resulting performance. The second contribution are optimized extraction and matching methods, exploiting the segment-based descriptor representation, and enabling real-time use cases.

The rest of this paper is organized as follows. Section II describes related work, including the descriptors we use in this work. We describe the temporal compression of different visual features in Section III, and the proposed optimizations in Section IV. Experimental results on a large video data set are reported for both temporal compression and optimizations in Section V. Section VI concludes the paper.

II. Related work

Related topics to our problem are instance search in video and near duplicate video detection. As mentioned above, copy detection approaches are not sufficiently robust against the content variations encountered in our setting.

One general way of speeding up processing is to make the feature extraction process for descriptors such as SIFT [2] more efficient. This can be done by more efficient and compact features (e.g., [3]), using GPUs (e.g., [4]) or dedicated hardware (e.g., [5], [6]) for feature extraction. As the feature extraction step for a single frame is treated as a black box in this paper, these approaches are complementary, and can be plugged into our framework.

Instance search, i.e., finding video clips containing a similar foreground object, background or scene as in the query, is still a challenging problem in large-scale video collections. In contrast to video copy detection, the problem cannot be addressed only by global visual descriptors, due to the variability with which the object of interest may be depicted. In recent years, there has been significant progress in defining more compact visual descriptors, typically by aggregating local descriptors (either sampled from interest points or densely) and applying means such as dimensionality reductions and binarization. Examples of such methods are Fisher Vectors [7], VLAD [8] and its improvements [9], VLAT [10] and MPEG CDVS [11]. While these descriptors achieve good matching performance even at small descriptor sizes, they are all descriptors for still images that need to be applied independently to individual frames of the video. Thus, they do not make use of the temporal redundancy of the video. This is not only an issue of the size of the extracted descriptor, but also of the matching complexity, as pairwise matching of the frame descriptors has to be performed.

Many approaches for finding near duplicate video are defined as a retrieval task, i.e., a database is established, including the creation of an efficient indexing data structure and then queried in real-time. If we consider use cases with continuous incoming content streams, the task is more one of near duplicate video detection. This distinction is also made in the state of the art survey in [12], and the authors observe that there are only few works dealing with near duplicate detection in a real-time scenario.

The method proposed in [13] assumes the existence of a shot structure and consistent luminance changes across matching streams, which does not generally hold, e.g. for user generated content (UGC). [14] proposes to exploit the redundancy within video segments to obtain a more compact description. Descriptors from interest points in each of the frames are matched against those of other frames in the segment, and redundant descriptors are removed, thus obtaining a more compact set of descriptors for the segment. An iterative method for matching near duplicate video segments is proposed in [15], using per frame ternary descriptors. However, the iterative process is used to optimize the complete alignment, but does not support early stopping in case of not or weakly matching segments. [16] propose a matching approach for videos based on spatio-temporal pyramid matching. While we share the basic approach of hierarchical matching with this approach, the differences are that they work on a set of raw descriptors that still have spatio-temporal association, rather than an already compacted and difference code data structure.

Deep convolutional neural networks (DNNs) have been very actively researched in recent years, in particular for image classification tasks. It has also been shown that the feature representations learned in the convolutional layers of the network (before performing the actual classification) are useful for many computer vision problems, and perform comparable or even better as established hand-crafted features [17]. This includes instance search, however, it was found that the features obtained from DNNs are less robust against geometric transformations of images as hand-crafted ones (cf. [18]). One strategy to address this issue is to add layers to the network, that perform pooling over transformed image patches. This approach, named nested invariance pooling (NIP) has been shown to address the robustness issue and provide significant performance improvements for instance search [19]. However, the results show that the best performance is achieved when combining the deep feature descriptor with a global descriptor using Scalable Compressed Fisher Vectors (SCFV) [20]. Recently, an approach for using features from intermediate CNN layers for near-duplicate video retrieval has been proposed [21], showing that the additionally preserved structural information improves matching performance.

A descriptor for image sequences, which encodes a set of consecutive and related frames (i.e., a segment such as a shot) as a single descriptor has been proposed in [1]. The descriptor is created from an aggregation of sets of local descriptors from each of the images, and contains an aggregation of global descriptors and a time and location indexed set of the extracted local descriptors. The proposed method can use compact still image descriptors (such as MPEG CDVS) as its basis. The descriptor extraction uses local descriptor extraction from interest points (and can thus benefit from the accelerated extraction methods described above) and a method for aggregation of such descriptors to a global descriptor, but is agnostic of the specific type of descriptor and aggregation method (as long as they fulfill certain properties). The descriptor extraction process can be parameterized for different descriptor bit rates. Depending on the bit rate, temporal subsampling and possibly lossy compression of local descriptors is applied. We use this framework as the basis of our work, and compare the compression of different local and global features along the temporal dimension. In addition, we propose efficient extraction and matching approaching for supporting real-time application of the descriptor.

III. Temporal compression of different visual features

In order to obtain compact descriptors for video segments, we aim at exploiting the temporal redundancy of key frames extracted from the same segment, and perform lossy or lossless compression of the global and local features extracted from these key frames. For all the features, segment boundary detection using matching of color histograms (with a predefined threshold) is performed as a first step, and subsequent frames with high similarity discarded. Note that the segments are not required to coincide with shots, but are delineated by strong visual changes. The aim is to obtain segments that are visually homogeneous for efficient representation, but is not necessarily a semantic structure of the content. If the video has been edited, the set of segment boundaries will contain the shot boundaries. However, if we deal with live streams there may be no edits in the stream. The segmentation is thus configured to rather oversegment the video in order to obtain homogeneous segments which can be efficiently described. We also sample key frames based on visual similarity. Thus we obtain an irregularly sampled sequence of key frames for each segment.

We compare two types of features extracted from these key frames.

1) CDVS: A CDVS descriptor contains a set of local SIFT descriptors [2] sampled around ALP interest points [11], which are quantized to a ternary representation (using mode 0 of CDVS descriptor extraction specification, which results in max. 300 key points). In addition, it contains an aggregated global descriptor, represented as a Scalable Compressed Fisher Vector (SCFV) [20] as a binary vector. We thus obtain a global binary descriptor and a local descriptor with ternary features and interest points for up to 300 interest points per frame.

2) NIP+SCVF: We extract a binary SCFV global descriptor as specified by CDVS, and combine it with deep features using the NIP descriptor as proposed in [19]. The network used to extract the deep features is the VGG-16 network [22], pretrained on the ImageNet data set, and adding the invariance

Feature	Representation	Compression
SCVF	binary vector per key frame	medoid descriptor and difference descriptors, with ABAC
CDVS local	ternary vectors for up to 300 interest points per key frame	vectors from medoid frame, filtered vectors from other frames, with ABAC
NIP	binary vector per key frame	medoid descriptor and difference descriptors, with ABAC

TABLE I
FEATURES, REPRESENTATION AND COMPRESSION APPROACH.

pooling layers. The resulting deep feature vector has a dimension of 512. It is linearized by subtracting a mean deep feature vector, and setting elements ≥ 0 to one and others to zero. The mean deep feature vector is determined from the distractor videos of the MPEG CDVA data set (see Section V-A). This configuration thus consists of two global (i.e., aggregated over a whole key frame) descriptors.

We apply the following compression strategies to the feature combinations.

1) CDVS: We apply lossless compression to the SCFV global descriptor. The descriptor of the medoid key frame of the segment is used as a reference, and the element-wise difference (XOR) between the SCFV descriptor of the medoid key frame and each of the other key frames in the segment is determined. The descriptor of the medoid frame and the difference descriptors are concatenated, and adaptive binary arithmetic coding (ABAC) [23] is applied. Similarly, all local features of the medoid key frame are added to a list. For the local features of all other frames, the distance to a feature already in the list is determined as the number of different elements between the ternary vectors. If the distance does not exceed a threshold θ_l, then the feature is discarded, and only a reference is kept, otherwise the feature is added to the list. Finally, the list of features is encoded using ABAC. Note that all features are encoded with their absolute values, as it was found that using difference descriptors does not improve the result.

2) NIP+SCVF: We use the same lossless encoding for the SCFV global descriptor as described above. For the deep features, we consider both lossless encoding (using the same approach as for SCFV) and lossy encoding. If lossy encoding is used, the deep feature descriptors for a key frame is replaced with a reference, if the Hamming distance to the descriptor of the medoid key frame does not exceed a threshold θ_d. Again, the sequence of the medoid descriptor and the remaining medoid descriptors are encoded using ABAC.

The different features, their representation and the compression approach are summarized in Table I.

IV. FAST EXTRACTION AND MATCHING

In this section we analyze the performance gains that can be achieved during extraction and matching based on a structured segment descriptor. The methods described in this section are agnostic of the specific feature, and could be implemented using any of the features described in the previous section. However, we follow [1] in basing the compact image sequence descriptor on the MPEG CDVS descriptor, making use of the global and local parts of the descriptor.

A. Extraction

Segmentation and key frame selection are performed as described in Section III. We then extract CDVS descriptors for each of the key frames. In order to represent the segment, we select the medoid of the set of key frames of each segment, i.e., the frame with a global descriptor with the minimal summed distance to the global descriptors of all other frames in the segment. This is a costly step, as a quadratic number of distance calculations is required. In [24], an approximation for extracting the medoid of time series has been proposed. While the approximation will often not find the true medoid, it has been shown that the use of the approximation does not have a negative impact when using it in visual matching and retrieval tasks. Once the approximate medoid is identified, the other global descriptors in the segment are stored in an order, that maximizes the information gain with each additional frame encoded. This means that the first frame after the medoid is chosen as the one most dissimilar to the medoid (i.e., one of the pivots in the approximation method), and each further frame is chosen to be dissimilar to the already encoded ones. This is a prerequisite for terminating matching early, when matching descriptors.

The local descriptors are encoded as one list for the entire segment, starting with those of the medoid frame. Then the encoding proceeds in both forward and backward temporal order from the medoid frame, and encodes only those local descriptors, that are more dissimilar than a threshold θ_l to the most similar descriptor already encoded. Otherwise, just a reference to the descriptor is stored.

B. Matching

We propose two matching methods, that take advantage of the medoid frame's descriptor as a representative for the segment, or use only a few of the frame descriptors encoded in the segment descriptor.

In optimized matching method 1, the global descriptors of the medoid frames of the two segments are compared. If the similarity exceeds a threshold θ_{med}, then full matching of global and local descriptors is performed. The threshold is determined as $\theta_{med} = \tau\theta_g$, where θ_g is the threshold used for matching global descriptors (depending on the descriptor configuration) and $0 \leq \tau \leq 1$ (i.e., $\tau = 0$ corresponds to the case of full exhaustive matching of frame descriptor pairs).

In optimized matching method 2, the global descriptors of the frames in each of the segment descriptors are compared incrementally in the order in which they are stored, which is starting with the medoid descriptor, and then followed by the descriptor which is most dissimilar to the preceding ones. The number of descriptors to be matched is determined as

$$N_{match} = 1 + \frac{1}{b}\min(seglen_A, seglen_B),$$

where $seglen_A$ is the number of frames encoded in segment descriptor A. The constant 1 ensures that at least the pair of medoid descriptors is matched. The result is identical to method 1 in cases where the minimum segment length is less than b.

If the similarity of any of the compared frame pairs exceeds a threshold θ_{med}, then full matching of global and local descriptors is performed. The threshold is determined as θ_{med}, defined as for method 1.

V. Results

A. Data set

We use the data set collected by MPEG for an activity called Compact Descriptors for Visual Analysis (CDVA) [25] in our experiments. The data set contains in total around 23,000 video clips with durations ranging from about one minute to more than an hour. The material contains broadcast and user generated content in different resolutions and frame rates, and with diverse contents. It is divided into a set of reference and query clips, which contain different views of one object or scene, embedded into noise clips. In addition, part of query clips have been modified with transformations (e.g., resolution and frame rate changes, overlays, screen capture). The rest of the set contains distractor material for retrieval experiments.

We perform pairwise matching of the 9,715 queries against the 5,128 reference clips, and report the true positive rate at 1% false positive rate and the Jaccard index of temporal localization of the matching segments. Further details on the data and the evaluation metrics can be found in [26].

B. Comparison of features

The results of applying compression are reported in Figure 1. One important observation is that the SCFV+NIP descriptor is significantly smaller, thus it is plotted separately. Despite the small size, lossless compression of both components halves the size of the SCFV+NIP descriptor. Additional lossy compression does not result in significant size reductions, but reduces the performance considerably. For the CDVS descriptor, lossless compression only results on about 5% size reduction. However, a significant amount of lossy compression can be applied with only small impact on the performance. Reducing the descriptor size to about one third of the original size reduces the performance by about 3%, only then the impact on the matching performance becomes larger.

We also analyze the performance of the features for different types of queries. The MPEG CDVA data set classifies queries into three categories: large objects (e.g., buildings), small objects (e.g. book) and scenes (e.g., interior of the same room, without a single large salient object). In addition, transformations are applied to part of the queries. Figure 2 shows the results for the different query types and the different features. One interesting observation is that the deep features always perform at least as good as the hand-crafted features. For small objects, which undergo stronger geometric transformations, the features perform equally well, while for scenes, where the notion of similarity is more fuzzy, the deep features strongly outperform the hand-crafted ones. Also for some transformations, in particular camcording, the performance difference is remarkable.

Method	τ	b	fraction matched		TPR	Jaccard
			matching	not matching		
full	-	-	1.000	1.000	0.845	0.651
opt 1	0.1	-	1.000	1.000	0.837	0.647
opt 1	0.3	-	0.990	0.985	0.835	0.647
opt 1	0.5	-	0.884	0.879	0.820	0.625
opt 1	0.7	-	0.636	0.588	0.769	0.612
opt 1	0.9	-	0.351	0.247	0.678	0.573
opt 2	0.3	2	1.000	1.000	0.836	0.629
opt 2	0.5	2	0.965	0.954	0.828	0.627
opt 2	0.7	2	0.942	0.918	0.815	0.624
opt 2	0.9	2	0.750	0.676	0.776	0.620
opt 2	1.0	2	0.607	0.503	0.748	0.610
opt 2	0.3	5	1.000	1.000	0.836	0.629
opt 2	0.5	5	0.965	0.954	0.828	0.627
opt 2	0.9	5	0.560	0.461	0.732	0.604

TABLE II
MATCHING PERFORMANCE OF OPTIMIZED MATCHING METHODS: TRUE POSITIVE RATE (TPR) AT 1% FALSE POSITIVE RATE AND JACCARD INDEX OF TEMPORAL LOCALIZATION. ALSO THE AVERAGE FRACTION OF ACTUALLY MATCHED GLOBAL DESCRIPTOR PAIRS FOR MATCHING AND NON-MATCHING PAIRS OF SEGMENT DESCRIPTORS IS SPECIFIED.

C. Fast matching

1) Matching performance: We measure the matching performance of matching all descriptors exhaustively (i.e., for all pairs of key frames encoded in the descriptors to be matched) and compare it with optimized matching approaches. Table II provides an overview of the results.

We can observe that for low values of τ (i.e., many descriptor pairs are matched) the differences between methods and parametrizations are small. However, there is in all cases a performance drop of at least 0.008 over exhaustive matching. As τ increases, the differences increase, and the benefit of matching more than just the medoid descriptor pair becomes visible. Similarly, for optimized method 2, there are no differences for different values of b up to $\tau = 0.5$. However, this increases to a performance gap of about 0.04 between $b = 2$ and $b = 5$ at $\tau = 0.9$.

Figure 3 shows the TPR vs. the fraction of frame descriptor pairs in each of the methods. The fraction of frame descriptor pairs matched is calculated as number of pairs matched in the selection step (e.g., 1 for optimized method 1) plus the frame descriptor pairs matched, if the threshold in the selection step is exceeded. The number is normalized with the number of frame descriptor pairs matched in a full exhaustive match, i.e. 1.0 corresponds to $|A| \times |B|$ for matching descriptors A and B, where $|A|$ is the number of frames encoded in descriptor A. We can see that there is roughly a linear relation, which allows choosing the tradeoff between decrease in matching performance and matching speed. A reduction of matching performance of about 5% reduces the number of frame descriptors pairs to be matched by 30%, and a decrease of about 10% halves the number of required matches.

2) Runtime: For runtime measurements, a machine with 2x Intel Xeon Processor E5-2630 v2, 2.6GHz (=2x 6 cores) with

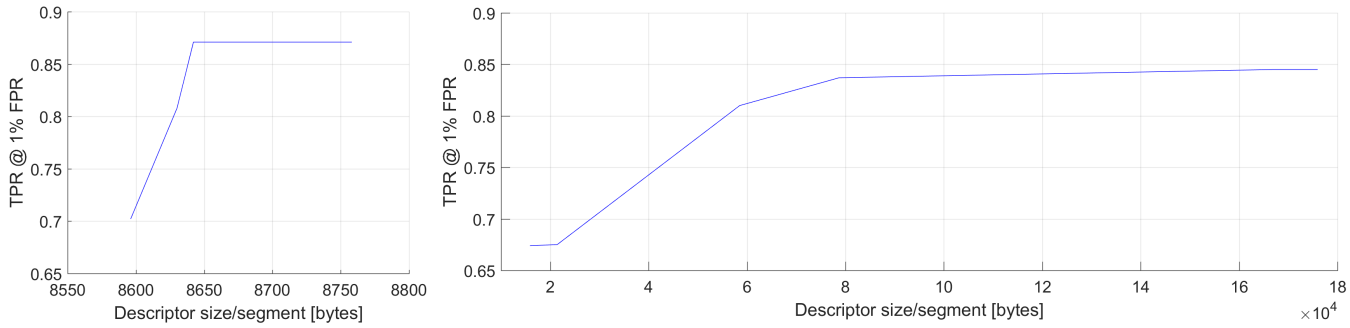

Fig. 1. Descriptor size vs. matching performance (true positive rate at 1% false positive rate) for different descriptors and compression (left: NIP+SCFV, right: CDVS).

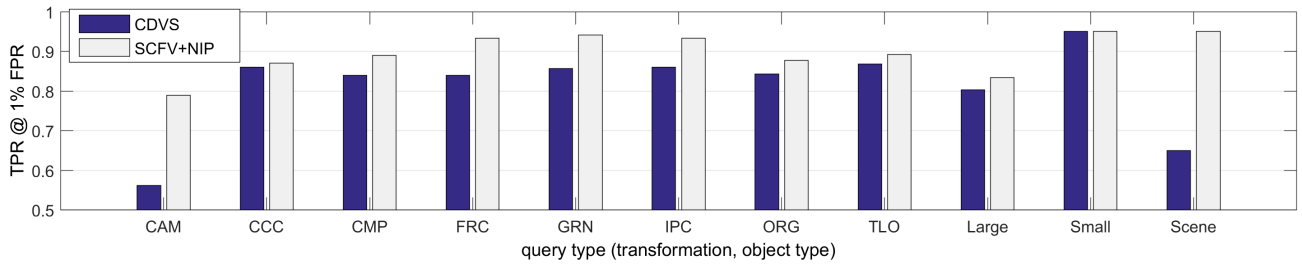

Fig. 2. Matching performance (true positive rate at 1% false positive rate) for different query types. Transformations: camcording (CAM), contrast/color change (CCC), transcoding (CMP), frame rate change (FRC), added film grain (GRN), interlaced/progressive conversion (IPC), unmodified (ORG), text/logo overlay (TLO).

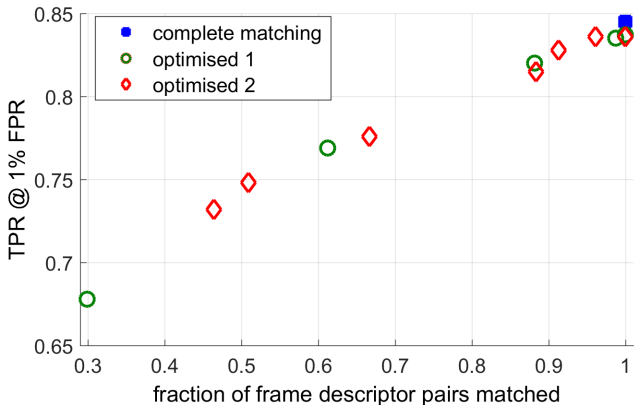

Fig. 3. Matching performance vs. fraction of descriptors matched.

128 GB RAM has been used. In accordance with the MPEG CDVA evaluation guidelines, the results are reported for single CPU core only, not using any GPU acceleration.

The average runtime for medoid calculation reduces from 2.572 ms per second of video for exact medoid calculation to 0.0148 ms per second of video for the approximate medoid calculation. Note that for segments with three key frames or less the exact medoid is determined in both cases. The total time needed for extraction, including I/O, video decoding and descriptor serialization is 0.69 s per second of video. This is due to the measurement on a file-based data set. When processing raw live streams, file I/O and decoding is not necessary, and thus the overall gain in terms of runtime performance is more visible.

For the optimized matching methods, the matching time per segment pair ranges from 29.19 ms for an average pair of non-matching segments for method 1 to 67.81 ms for an average pair of matching segments for method 2. Assuming 50 ms as average matching time, we can match 20 segment pairs per second on one CPU core. If we assume a segment duration of 10s (which rather corresponds to edited content, and is a very cautious assumption for UGC), this means that a segment of a stream can be matched against the content from the past five minutes from 6 streams on a single core in real-time. As the extraction and matching tasks are well suited for parallelization, the use of multiple cores or GPUs can provide significant speedup.

VI. Conclusion

In order to enable efficient instance search in video, compact descriptors for video segments have been proposed. They exploit the temporal redundancy within a video segment to obtain smaller descriptors, and the segment-based representation can be exploited to enable more efficient matching. In this paper we analyze the performance of different visual features when applying both lossless and lossy compression to the set of descriptors of one video segment. We consider both hand-crafted and deep features, i.e., visual features obtained from training a deep convolutional neural network. We also propose optimizations to the extraction and matching procedure,

which can be applied to different visual features. Both the compression methods and the optimizations are experimentally evaluated on a large video data set.

We have proposed the use of an optimized method for medoid calculation in the descriptor extraction for segment-based video descriptors, as well as optimized matching methods that make use of the medoid as a representative of the extracted data. We provide experimental results on a large data set, showing that real-time extraction and matching of descriptors from an incoming stream and matching against recent data from other streams is even feasible on a single core per stream.

We have also compared the performance of hand-crafted and deep features for different types of queries and transformations, and analyzed how temporal compression can be applied to each of the descriptor types. Deep features perform always at least as good as hand-crafted ones. While hand-crafted features can be significantly reduced using lossy compression, the deep features are already much smaller, allowing for about 25% reduction using lossless compression.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 732461, ReCAP ("Real-time Content Analysis and Processing", http://recap-project.com) and no. 761802, MARCONI ("Multimedia and Augmented Radio Creation: Online, iNteractive, Individual").

REFERENCES

[1] W. Bailer, S. Wechtitsch, and M. Thaler, "Compressing visual descriptors of image sequences," in *Proceedings of the 23rd International Conference MultiMedia Modeling*, Reykjavik, IS, Jan. 2017.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[4] H. Fassold and J. Rosner, "A real-time gpu implementation of the sift algorithm for large-scale video analysis tasks," in *SPIE/IS&T Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 940 007–940 007.

[5] F.-C. Huang, S.-Y. Huang, J.-W. Ker, and Y.-C. Chen, "High-performance sift hardware accelerator for real-time image feature extraction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 340–351, 2012.

[6] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y.-C. Chang, "Fast sift design for real-time visual feature extraction," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3158–3167, 2013.

[7] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE Conf. Computer Vision and Pattern Recognition*, June 2007.

[8] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3304–3311.

[9] R. Arandjelovic and A. Zisserman, "All about vlad," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1578–1585.

[10] D. Picard and P.-H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *IEEE International Conference on Image Processing*, Brussels, BE, Sept. 2011.

[11] "ISO/IEC 15938-13, Information technology – Multimedia content description interface – Part 13: Compact descriptors for visual search," 2015.

[12] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013.

[13] Q. Xie, Z. Huang, H. T. Shen, X. Zhou, and C. Pang, "Efficient and continuous near-duplicate video detection," in *Web Conference (APWEB), 2010 12th International Asia-Pacific*. IEEE, 2010, pp. 260–266.

[14] X. Zhou, X. Zhou, L. Chen, A. Bouguettaya, N. Xiao, and J. A. Taylor, "An efficient near-duplicate video shot detection method using shot-based interest points," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 879–891, 2009.

[15] K.-R. Kim, W.-D. Jang, and C.-S. Kim, "Frame-level matching of near duplicate videos based on ternary frame descriptor and iterative refinement," in *IEEE International Conference on Image Processing*, 2015, pp. 31–35.

[16] J. Choi, W. J. Jeon, and S.-C. Lee, "Spatio-temporal pyramid matching for sports videos," in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. ACM, 2008, pp. 291–297.

[17] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[18] O. Morère, J. Lin, A. Veillard, L.-Y. Duan, V. Chandrasekhar, and T. Poggio, "Nested invariance pooling and rbm hashing for image instance retrieval," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 260–268.

[19] Y. Lou, Y. Bai, J. Lin, S. Wang, J. Chen, V. Chandrasekhar, L. Y. Duan, T. Huang, A. C. Kot, and W. Gao, "Compact deep invariant descriptors for video retrieval," in *Data Compression Conference (DCC)*, April 2017, pp. 420–429.

[20] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao, "Rate-adaptive compact fisher codes for mobile visual search," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 195–198, 2014.

[21] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval by aggregating intermediate CNN layers," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 251–263.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] G. Langdon, "Adaptive binary arithmetic coding for multi-media applications," in *Compcon Spring'91. Digest of Papers*, 1991, pp. 354–357.

[24] W. Bailer, M. Winter, and S. Wechtitsch, "Learning selection of user generated event videos," in *Workshop on Content-based Multimedia Indexing*, Firenze, IT, Jun. 2017.

[25] "Call for Proposals for Compact Descriptors for Video Analysis (CDVA) – Search and Retrieval," Tech. Rep. ISO/IEC JTC1/SC29/WG11/N15339, 2015.

[26] "Evaluation framework for compact descriptors for video analysis – search and retrieval – version 2.0," Tech. Rep. ISO/IEC JTC1/SC29/WG11/N15729, 2015.